Table of Contents

🏗️ HealthPal Backend — Project Structure Overview

> ⚠️ **Technology Stack:** Node.js (Express.js) + MySQL (via Sequelize) **Team Members:**
>
> - 🧑‍⚕️ **Yousef Hannani** – Remote Consultations & Authentication
> - 💰 **Yazan Jamal** – Sponsorships / Donations & Security Middleware
> - 💊 **Hazem Mahameed** – Medication & Equipment Coordination / Logging
> - 🧠 **Muath Hassoun** – Health Education & Mental Support / Documentation

```
healthpal-backend/
├── src/
│   ├── config/
│   │   ├── db.js
│   │   ├── logger.js
│   │   └── env.js
│   │
│   ├── models/
│   │   ├── user.model.js
│   │   ├── consultation.model.js
│   │   ├── donation.model.js
│   │   ├── medication.model.js
│   │   ├── education.model.js
│   │   └── index.js
│   │
│   ├── routes/
│   │   ├── user.routes.js
│   │   ├── consultation.routes.js
│   │   ├── donation.routes.js
│   │   ├── medication.routes.js
│   │   ├── education.routes.js
│   │   └── index.js
│   │
│   ├── controllers/
│   │   ├── user.controller.js
│   │   ├── consultation.controller.js
│   │   ├── donation.controller.js
│   │   ├── medication.controller.js
│   │   ├── education.controller.js
│   │   └── index.js
│   │
│   ├── services/
│   │   ├── user.service.js
│   │   ├── consultation.service.js
│   │   ├── donation.service.js
│   │   ├── medication.service.js
│   │   ├── education.service.js
│   │   └── translation.service.js
│   │
│   ├── middleware/
│   │   ├── auth.middleware.js
│   │   ├── error.middleware.js
│   │   ├── validation.middleware.js
│   │   └── role.middleware.js
│   │
│   ├── utils/
│   │   ├── apiResponse.js
│   │   ├── emailHelper.js
│   │   ├── paymentHelper.js
│   │   ├── matchingHelper.js
│   │   └── notificationHelper.js
│   │
│   ├── logs/
│   │   └── (runtime logs)
│   │
│   ├── app.js
│   └── server.js
│
```

```
├── tests/
│   ├── unit/
│   ├── integration/
│   └── postman_collection.json
│
├── public/
│   ├── uploads/
│   └── static/
│
├── docs/
│   ├── API_Documentation.md
│   ├── Database_Schema.png
│   └── System_Architecture.pdf
│
├── .env
├── .gitignore
├── package.json
├── package-lock.json
└── README.md
```

📁 src/ — Core Application Source

This folder contains all the backend logic, including configuration, API routes, controllers, services, and utilities.

⚙️ src/config/

Purpose: System-wide configuration files. **Responsible:** *Shared setup (Yousef & Hazem)*

File	Description
`db.js`	Database connection file using Sequelize (MySQL).
`logger.js`	Winston or custom logger setup (Hazem manages runtime logs).
`env.js`	Loads environment variables using `dotenv`.

🧩 src/models/

Purpose: Defines database models (tables) and their relationships. **Responsible:** *Each member for their own module's models.*

File	Owner	Description
`user.model.js`	Yousef	User schema (patients, doctors, donors, NGOs) with role field.
`consultation.model.js`	Yousef	Manages appointment sessions, types (video/chat), and status.
`donation.model.js`	Yazan	Represents donation campaigns, sponsors, and transactions.
`medication.model.js`	Hazem	Stores medicine/equipment inventory and availability.
`education.model.js`	Muath	Articles, workshops, and mental support resources.
`index.js`	Shared	Combines and exports all Sequelize models.

🌐 src/routes/

Purpose: Defines all RESTful API endpoints and connects them to controllers. **Responsible:** *Each member handles their module routes.*

File	Owner	Description
`user.routes.js`	Yousef	Handles authentication, registration, and role-based routes.
`consultation.routes.js`	Yousef	API routes for remote consultation sessions.
`donation.routes.js`	Yazan	Routes for donations, campaigns, and invoices.
`medication.routes.js`	Hazem	Routes for managing medicine and equipment coordination.
`education.routes.js`	Muath	Routes for educational content and mental support.
`index.js`	Shared	Combines all routers into one main router.

🧠 src/controllers/

Purpose: Controls the business logic for each route. **Responsible:** *Each member writes logic for their module.*

File	Owner	Description
`user.controller.js`	Yousef	Handles login, signup, JWT tokens, and profile management.
`consultation.controller.js`	Yousef	Manages consultations (CRUD + scheduling).
`donation.controller.js`	Yazan	Handles donations, payments, and campaign progress.
`medication.controller.js`	Hazem	Manages inventory, matching, and availability tracking.
`education.controller.js`	Muath	Controls articles, guides, and mental health resources.
`index.js`	Shared	Exports all controllers.

🔧 src/services/

Purpose: Business-level logic and integrations (APIs, helpers, or third-party).
Responsible: *Each member contributes based on module requirements.*

File	Owner	Description
`user.service.js`	Yousef	User operations and authentication helpers.
`consultation.service.js`	Yousef	Session scheduling and translation handling.
`donation.service.js`	Yazan	Donation management and payment service integration.
`medication.service.js`	Hazem	Handles medicine/equipment logic and matching algorithm.
`education.service.js`	Muath	Manages content retrieval and caching.
`translation.service.js`	Yousef	Integrates Google Translate API for bilingual consultations.

🧱 src/middleware/

Purpose: Middleware for validation, authentication, error handling, and roles.
Responsible: *Shared setup with key owners.*

File	Owner	Description
`auth.middleware.js`	Yousef	Verifies JWT tokens and user sessions.
`error.middleware.js`	Hazem	Global error handling and logging.
`validation.middleware.js`	Yazan	Validates incoming request data using Joi.
`role.middleware.js`	Yousef	Checks user roles and permissions.

🧰 src/utils/

Purpose: Helper functions and reusable utility modules. **Responsible:** *Shared among all.*

File	Owner	Description
`apiResponse.js`	Shared	Standardizes API response formats.
`emailHelper.js`	Muath	Sends email notifications for events (support, campaigns).
`paymentHelper.js`	Yazan	Simulates or integrates donation payment flow.
`matchingHelper.js`	Hazem	Matches patients' needs with available equipment.
`notificationHelper.js`	Yousef	Sends reminders or push notifications.

🪵 src/logs/

Purpose: Stores auto-generated runtime logs (via Winston or custom logger). **Responsible:** *Hazem Mahameed*

> ⚠ Log files will include error logs, request tracking, and DB activity traces.

🚀 Core Files

File	Description
`app.js`	Initializes Express, middleware, routes, and error handlers.
`server.js`	Starts the server and connects to the database.

🧪 tests/

Purpose: Unit and integration testing. **Responsible:** *Everyone writes tests for their modules.*

Folder/File	Description
`unit/`	Unit tests for services, controllers, and utils.
`integration/`	Full API integration tests.
`postman_collection.json`	Postman collection for manual API testing.

🌍 public/

Purpose: Static assets (uploads, reports, static files). **Responsible:** *Shared.*

Folder	Description
`uploads/`	Patient reports, medical documents, etc.
`static/`	Optional public assets or frontend placeholders.

📘 docs/

Purpose: Documentation for APIs, database, and architecture. **Responsible:** *Muath Hassoun.*

File	Description
`API_Documentation.md`	API endpoints documentation (Swagger/Postman).
`Database_Schema.png`	Database ER diagram.
`System_Architecture.pdf`	System architecture overview.

🧩 Root Files

File	Description
`.env`	Environment variables (DB credentials, JWT secrets, etc.).
`.gitignore`	Files and folders ignored by Git.
`package.json`	Node project dependencies and scripts.
`package-lock.json`	Auto-generated dependency lock file.
`README.md`	General project overview and setup instructions.

✅ Summary of Responsibility

Member	Main Modules	Shared Responsibilities
Yousef Hannani	Users, Consultations	Authentication, Role middleware
Yazan Jamal	Donations, Sponsorships	Validation, Security
Hazem Mahameed	Medication, Equipment	Logging, Error handling
Muath Hassoun	Education, Mental Health	Documentation, Email notifications

> ⚠ 💡 **Tip:** Keep your code modular and consistent with naming conventions. Use `.service.js` for reusable business logic and `.controller.js` only for route-level logic.